

Session Code: TLS200

tools & languages

Visual Basic “Whidbey”: RAD for the Visual Basic Developer

Jay Schmelzer and Shamez Rajan
Program Manager
Microsoft Corporation

PDC⁰³

Make the connection

Microsoft[®]

Agenda

- Visual Basic “Whidbey” design goals
- Writing correct code
- Improving the debugging cycle
- Common tasks made easier

Visual Basic Initiatives

- Increase productivity by cutting development time in half and reducing the amount of code to write for common scenarios by more than 50%
- Leverage the full power of Windows desktop to create rich windows client applications with the deployment and update capabilities of the web
- Continue to provide full access to the platform and runtime features

Increasing Productivity

- Do more with less code and fewer clicks
 - My provides quick access to common functionality
 - Smart tasks, snap lines, in place property editing
 - Data schema used to rapidly create applications
 - Web Form themes, templates, and site personalization
- Revitalize the iterative development experience
 - Faster and streamlined development environment
 - More efficient interactive debugging cycle
 - Rapidly code solutions with Snippets and AutoCorrect
 - Task oriented MyHelp targeted for VB developers

The Power of Windows

- ClickOnce Deployment
 - Publish, maintain rich client apps with auto update
 - Simplified programming in the sandbox
- Richer, modern, compelling user interfaces
 - Automatic XP Theme support
 - Office look and feel with ToolStrip, MenuStrip controls
 - Create more responsive apps with background tasks
- Zero code UI and data tiers
 - Drag and drop schema, web services, business objects

Full Access to the Platform

- Powerful language features
 - Operator overloading
 - Unsigned types
 - Partial types
- Full Generics support
 - Creation and consumption
 - Improved compile time code validation
- Full support for Yukon managed procedures
- Full support for MSBuild

Writing Correct Code

- Design-time expression evaluation
 - Explore referenced assemblies
 - Test the code you write
- Error correction
 - Smart tag based implementation similar to Office spelling and grammar check
 - Suggests fixes for common syntax errors

Improved Debugging Cycle

- Change code without stopping / re-starting your debugging session
 - Fix the issue while code is executing under the debugger
 - Visual indication of “legal” vs. “illegal” edits
- Access to debugger values
 - Debugger visualizers
 - Enhanced data tool tips
 - Formatted debugger output windows

Improved Debugging Cycle

- Improved exception interaction
 - Ability to continue after an exception
 - Break in the debugger on exception states (First Chance, Unhandled and User-unhandled)
- New Exception Assistant
 - Provides access to exception details
 - Includes troubleshooting tips for handling the exception

Writing Correct Code

demo

Common Tasks Made Easier

● Code Snippets

Part I - Editor Features

- Reusable snippets of code
- Proactively prompt for customizations
- Includes required project references and file imports
- ~500 task oriented snippets included

● Extensible Architecture

- Specify multiple snippet stores (local, network share, etc.)
- Editor to create new snippet or modify existing snippets

Code Snippets

demo

My.

A Speed Dial into the Framework

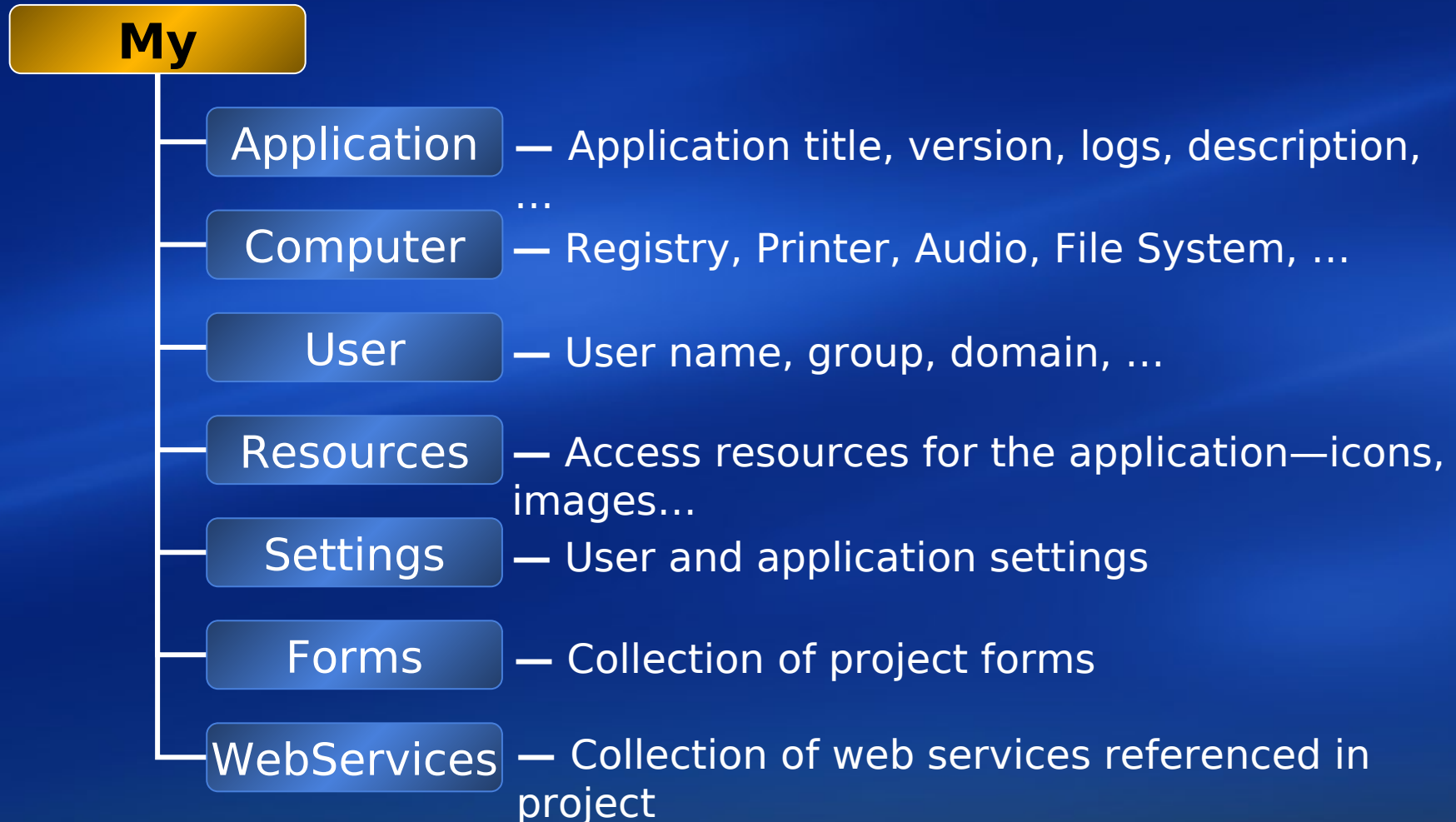
- My.Application
 - Version Information, Log, Startup and Shutdown events, Opened Forms, Instance Style, ClickOnce Deployment API, ...
- My.Computer
 - Audio, Printer, Screen, Keyboard, Mouse, Registry, File System, Serial Ports, Network, ...
- My.User
 - User Login / Domain Name, Display Name, Roles, ...

My.

An Application Level Version of Me

- My.Forms
 - `My.Forms.Form1.Show`
- My.WebServices
 - `My.WebServices.MSDN.Search("VB")`
- My.Resources
 - `PictureBox1.Image =
My.Resources.Logo`
- My.Settings
 - `My.Settings.User.FormLocation =
Me.Location`

My Object Hierarchy



My.Application

My.Application is designed to:

- Make application-related properties more accessible
- Allow commonly referenced services and classes to be enabled easily
- Provide a more manageable framework for application startup and shutdown.

My.Application



My.Computer

My.Computer provides straightforward access to the host computer's properties and hardware resources, allowing devices to be enabled easily and their services integrated seamlessly into applications.

My.Computer



```
graph TD; Root[My.Computer] --- Mouse[Mouse]; Root --- Audio[Audio]; Root --- Network[Network]; Root --- Printers[Printers]; Root --- Registry[Registry]; Root --- Clipboard[Clipboard]; Root --- FileSystem[FileSystem]; Root --- Ellipsis[...];
```

Mouse

Audio

Network

Printers

Registry

Clipboard

FileSystem

...

Using My. demo

PDC⁰³

Make the connection

Summary

- Visual Basic “Whidbey” improves framework productivity
 - Improves navigation of the Framework
 - 50% less code for common tasks
- Visual Basic “Whidbey” enables building robust applications in less time
 - Identify and fix errors before you build
 - Provides the most productive Visual Basic debugging environment yet!

Additional Resources

- Related sessions

- TLS300 – Visual Basic “Whidbey”: Advanced Language and IDE Features (Tuesday 3:45 Room 402 AB)
- TLS343 – Visual Studio “Whidbey”: Advanced Debugging Techniques (Tuesday 5:15 Room 408 AB)
- TLS344 – Visual Studio “Whidbey”: Deploying Applications Using ClickOnce (Wednesday 10:00 Room 511 ABC)
- ARC413 – CLR Under the Covers: “Whidbey” CLR Internals (Wednesday 5:00 Room 152/153)

- Send us feedback!

- vswish@microsoft.com



PDC⁰³

Make the connection

Microsoft Professional Developers Conference 2003

October 26 - 30, 2003, Los Angeles, CA

Microsoft®